Audio coding

The invention relates to coding at least part of an audio signal.

In the art of audio coding, Linear Predictive Coding (LPC) is well known for representing spectral content. Further, many efficient quantization schemes have been proposed for such linear predictive systems, e.g. Log Area Ratios [1], Reflection Coefficients [2] and Line Spectral Representations such as Line Spectral Pairs or Line Spectral Frequencies [3, 4, 5].

Without going into much detail on how the filter-coefficients are transformed to a Line Spectral Representation (reference is made to [6, 7, 8, 9, 10] for more detail), the results are that an M-th order all-pole LPC filter H(z) is transformed to M frequencies, often referred to as Line Spectral Frequencies (LSF). These frequencies uniquely represent the filter *H(z)*. As an example see Fig. 1. Note that for clarity the Line Spectral Frequencies have been depicted in Fig. 1 as lines towards the amplitude response of the filter, although they are nothing more than just frequencies, and thus do not in themselves contain any amplitude information whatsoever.

An object of the invention is to provide advantageous coding of at least part of an audio signal. To this end, the invention provides a method of encoding, an encoder, an encoded audio signal, a storage medium, a method of decoding, a decoder, a transmitter, a receiver and a system as defined in the independent claims. Advantageous embodiments are defined in the dependent claims.

According to a first aspect of the invention, at least part of an audio signal is coded in order to obtain an encoded signal, the coding comprising predictive coding the at least part of the audio signal in order to obtain prediction coefficients which represent temporal properties, such as a temporal envelope, of the at least part of the audio signal, transforming the prediction coefficients into a set of times representing the prediction coefficients, and including the set of times in the encoded signal. Note that times without any amplitude information suffice to represent the prediction coefficients.

Although a temporal shape of a signal or a component thereof can also be directly encoded in the form of a set of amplitude or gain values, it has been the inventor's insight that higher quality can be obtained by using predictive coding to obtain prediction coefficients which represent temporal properties such as a temporal envelope and transforming these prediction coefficients to into a set of times. Higher quality can be obtained because locally (where needed) higher time resolution can be obtained compared to fixed time-axis technique. The predictive coding may be implemented by using the amplitude response of an LPC filter to represent the temporal envelope.

It has been a further insight of the inventors that especially the use of a time domain derivative or equivalent of the Line Spectral Representation is advantageous in coding such prediction coefficients representing temporal envelopes, because with this technique times or time instants are well defined which makes them more suitable for further encoding. Therefore, with this aspect of the invention, an efficient coding of temporal properties of at least part of an audio signal is obtained, attributing to a better compression of the at least part of an audio signal.

Embodiments of the invention can be interpreted as using an LPC spectrum to describe a temporal envelope instead of a spectral envelope and that what is time in the case of a spectral envelope, now is frequency and vice versa, as shown in the bottom part of Fig. 2. This means that using a Line Spectral Representation now results in a set of times or time instances instead of frequencies. Note that in this approach times are not fixed at predetermined intervals on the time-axis, but that the times themselves represent the prediction coefficients.

The inventors realized that when using overlapping frame analysis/synthesis for the temporal envelope, redundancy in the Line Spectral Representation at the overlap can be exploited. Embodiments of the invention exploit this redundancy in an advantageous manner.

The invention and embodiments thereof are in particular advantageous for the coding of a temporal envelope of a noise component in the audio signal in a parametric audio coding schemes such as disclosed in WO 01/69593-A1. In such a parametric audio coding scheme, an audio signal may be dissected into transient signal components, sinusoidal signal components and noise components. The parameters representing the sinusoidal components may be amplitude, frequency and phase. For the transient components the extension of such parameters with an envelope description is an efficient representation.

Note that the invention and embodiments thereof can be applied to the entire relevant frequency band of the audio signal or a component thereof, but also to a smaller frequency band.

These and other aspects of the invention will be apparent from the elucidated with reference to the accompanying drawings.

In the drawings:

Fig. 1 shows an example of an LPC spectrum with 8 poles with corresponding 8 Line Spectral Frequencies according to prior art;

Fig. 2 shows (top) using LPC such that H(z) represents a frequency spectrum, (bottom) using LPC such that H(z) represents a temporal envelope;

Fig. 3 shows a stylized view of exemplary analysis/synthesis windowing;

Fig. 4 shows an example sequence of LSF times for two subsequent frames;

Fig. 5 shows matching of LSF times by shifting LSF times in a frame *k* relative to a previous frame *k-1*;

Fig. 6 shows weighting functions as function of overlap; and

Fig. 7 shows a system according to an embodiment of the invention.

The drawings only show those elements that are necessary to understand the embodiments of the invention.

Although the below description is directed to the use of an LPC filter and the calculation of time domain derivatives or equivalents of LSFs, the invention is also applicable to other filters and representations which fall within the scope of the claims.

Fig. 2 shows how a predictive filter such as an LPC filter can be used to describe a temporal envelope of an audio signal or a component thereof. In order to be able to use a conventional LPC filter, the input signal is first transformed from time domain to frequency domain by e.g. a Fourier Transform. So in fact, the temporal shape is transformed in a spectral shape which is coded by a subsequent conventional LPC filter which is normally used to code a spectral shape. The LPC filter analysis provides prediction coefficients which represent the temporal shape of the input signal. There is a trade-off between time-resolution and frequency resolution. Say that e.g. the LPC spectrum would consist of a number of very sharp peaks (sinusoids). Then the auditory system is less sensitive to time-resolution changes, thus less resolution is needed, also the other way around, e.g. within a transient the resolution of the frequency spectrum does not need to be accurate. In this sense one could see this as a

combined coding, the resolution of the time-domain is dependent on the resolution of the frequency domain and vice versa. One could also employ multiple LPC curves for the time-domain estimation, e.g. a low and a high frequency band, also here the resolution could be dependent on the resolution of the frequency estimation etc, this could thus be exploited.

An LPC filter *H(z)* can generally be described as:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_m z^{-m}}$$

The coefficients a_i,with i running from 1 to m, are the prediction filter coefficients resulting from the LPC analysis. The coefficients a_i determine *H(z)*.

To calculate the time domain equivalents of the LSFs, the following procedure can be used. Most of this procedure is valid for a general all-pole filter H(z), so also for frequency domain. Other procedures known for deriving LSFs in the frequency domain can also be used to calculate the time domain equivalents of the LSFs.

The polynomial *A(z)* is split into two polynomials *P(z)* and *Q(z)* of order *m+1*. The polynomial *P(z)* is formed by adding a reflection coefficient (in lattice filter form) of *+1* to *A(z)*, *Q(z)* is formed by adding a reflection coefficient of *–1*. There's a recurrent relation between the LPC filter in the direct form (equation above) and the lattice form:

$$A_i(z) = A_{i-1}(z) + k_i z^{-i} A_{i-1}(z^{-1})$$

with $i=1,2,\ldots,m$, $A_0(z)=1$ and k_i the reflection coefficient.

The polynomials *P(z)* and *Q(z)* are obtained by:

$$P(z) = A_m(z) + z^{-(m+1)} A_m(z^{-1})$$

$$Q(z) = A_m(z) - z^{-(m+1)} A_m(z^{-1})$$

The polynomials $P(z) = 1 + p_1 z^{-1} + p_2 z^{-2} + \ldots + p_m z^{-m} + z^{-(m+1)}$ and $Q(z) = 1 + q_1 z^{-1} + q_2 z^{-2} + \ldots + q_m z^{-m} - z^{-(m+1)}$ obtained in this way are even symmetrical and anti-symmetrical:

$p_1 = p_m$ $\quad$ $q_1 = -q_m$

$p_2 = p_{m-1}$ $\quad$ $q_2 = -q_{m-1}$

. .

. .

Some important properties of these polynomials:

– All zeros of *P(z)* and *Q(z)* are on the unit circle in the z-plane.

- The zeros of *P(z)* and *Q(z)* are interlaced on the unit circle and do not overlap.
- Minimum phase property of *A(z)* is preserved after quantization guaranteeing stability of *H(z)*.

Both polynomials *P(z)* and *Q(z)* have *m+1* zeros. It can be easily seen that $z=-1$ and $z=1$ are always a zero in *P(z)* or *Q(z)*. Therefore they can be removed by dividing by $1+z^{-1}$ and $1-z^{-1}$. If *m* is even this leads to:

$$P'(z) = \frac{P(z)}{1+z^{-1}}$$

$$Q'(z) = \frac{Q(z)}{1-z^{-1}}$$

If *m* is odd:

$$P'(z) = P(z)$$

$$Q'(z) = \frac{Q(z)}{(1-z^{-1})(1+z^{-1})}$$

The zeros of the polynomials *P'(z)* and *Q'(z)* are now described by $z_i=e^{jt}$ because the LPC filter is applied in the temporal domain. The zeros of the polynomials *P'(z)* and *Q'(z)* are thus fully characterized by their time *t*, which runs from 0 to π over a frame, wherein 0 corresponds to a start of the frame and π to an end of that frame, which frame can actually have any practical length, e.g. 10 or 20 ms. The times *t* resulting from this derivation can be interpreted as time domain equivalents of the line spectral frequencies, which times are further called LSF times herein. To calculate the actual LSF times, the roots of *P'(z)* and *Q'(z)* have to be calculated. The different techniques that have been proposed in [9],[10],[11] can also be used in the present context.

Fig. 3 shows a stylized view of an exemplary situation for analysis and synthesis of temporal envelopes. At each frame *k* a, not necessarily rectangular, window is used to analyze the segment by LPC. So for each frame, after conversion, a set of *N* LSF times is obtained. Note that *N* in principal does not need to be constant, although in many cases this leads to a more efficient representation. In this embodiment we assume that the LSF times are uniformly quantized, although other techniques like vector quantization could also be applied here.

Experiments have shown that in an overlap area as shown in Fig. 3 there is often redundancy between the LSF times of frame *k-1* with those of frame *k*. Reference is also made to Figs. 4 and 5. In embodiments of the invention which are described below, this

redundancy is exploited to more efficiently encode the LSF times, which helps to better compress the at least part of an audio signal. Note that Figs. 4 and 5 show usual cases wherein the LSF times of frame *k* in the overlapping area are not identical but however rather close to the LSF times in frame *k-1*.

First embodiment using overlapping frames

In a first embodiment using overlapping frames it is assumed that the differences between LSF times of overlapping areas can be, perceptually, neglected or result in an acceptable loss in quality. For a pair of LSF times, one in the frame *k-1* and one in the frame k, a derived LSF time is derived which is a weighted average of the LSF times in the pair. A weighted average in this application is to be construed as including the case where only one out of the pair of LSF times is selected. Such a selection can be interpreted as a weighted average wherein the weight of the selected LSF time is one and the weight of the non-selected time is zero. It is also possible that both LSF times of the pair have the same weight.

For example, assume LSF times $\{ l_0, l_1, l_2, \ldots, l_N \}$ for frame *k-1* and $\{ l_0, l_1, l_2, \ldots, l_M \}$ for frame *k* as shown in Fig. 4. The LSF times in frame *k* are shifted such that a certain quantization level *l* is in the same position in each of the two frames. Now assume that there are three LSF times in the overlapping area for each frame, as is the case for Fig. 4 and Fig. 5. Then the following corresponding pairs can be formed: $\{l_{N-2,k-1}\ l_{0,k}\ ,\ l_{N-1,k-1}\ l_{1,k}\ ,\ l_{N,k-1}\ l_{2,k}\}$. In this embodiment, a new set of three derived LSF times is constructed based on the two original sets of three LSF times. A practical approach is to just take the LSF times of frame *k-1* (or *k*), and calculate the LSF times of frame *k* (or *k-1*) by simply shifting the LSF times of frame *k-1* (or *k*) to align the frames in time. This shifting is performed in both the encoder *and* the decoder. In the encoder the LSFs of the right frame *k* are shifted to match the ones in the left frame *k-1*. This is necessary to look for pairs and eventually determine the weighted average.

In preferred embodiments, the derived time or weighted average is encoded into the bit-stream as a 'representation level' which is an integer value e.g. from 0 until 255 (8 bits) representing 0 until pi. In practical embodiments also Huffman coding is applied. For a first frame the first LSF time is coded absolutely (no reference point), all subsequent LSF times (including the weighted ones at the end) are coded differentially to their predecessor. Now, say frame *k* could make use of the 'trick' using the last 3 LSF times of frame *k-1*. For decoding, frame *k* then takes the last three representation levels of frame *k-1* (which are at the

end of the region 0 until 255) and shift them back to its own time-axis (at the beginning of the region 0 until 255). All subsequent LSF times in frame k would be encoded differentially to their predecessor starting with the representation level (on the axis of frame k) corresponding to the last LSF in the overlap area. In case frame k could not make use of the 'trick' the first LSF time of frame k would be coded absolutely and all subsequent LSF times of frame k differential to their predecessor.

A practical approach is to take averages of each pair of corresponding LSF times, e.g. $(l_{N-2,k-1} + l_{0,k})/2$,$(l_{N-1,k-1} + l_{1,k})/2$ and $(l_{N,k-1} + l_{2,k})/2$.

An even more advantageous approach takes into account that the windows typically show a fade-in/fade-out behavior as shown in Fig. 3. In this approach a weighted mean of each pair is calculated which gives perceptually better results. The procedure for this is as follows. The overlapping area corresponds to the area (π-r, π). Weight functions are derived as depicted in Fig. 6. The weight to the times of the left frame *k-1* for each pair separately is calculated as:

$$w_{k-1} = \frac{\pi - l_{mean}}{r}$$

where l_{mean} is the mean (average) of a pair, e.g.: $l_{mean} =(l_{N-2,k-1} + l_{0,k}) / 2$.
The weight for frame *k* is calculated as $w_k=1-w_{k-1}$.
The new LSF times are now calculated as:

$$l_{weighted} = l_{k-1} w_{k-1} + l_k w_k$$

where l_{k-1} and l_k form a pair. Finally the weighted LSF times are uniformly quantized.

As the first frame in a bit-stream has no history, the first frame of LSF times always need to be coded without exploitation of techniques as mentioned above. This may be done by coding the first LSF time absolutely using Huffman coding, and all subsequent values differentially to their predecessor within a frame using a fixed Huffman table. All frames subsequent to the first frame can in essence make advantage of an above technique. Of course such a technique is not always advantageous. Think for instance of a situation where there are an equal number of LSF times in the overlap area for both frames, but with a very bad match. Calculating a (weighted) mean might then result in perceptual deterioration. Also the situation where in frame *k-1* the number of LSF times is not equal to the number of LSF times in frame *k* is preferably not defined by an above technique. Therefore for each frame of LSF times an indication, such as a single bit, is included in the encoded signal to indicate whether or not an above technique is used, i.e. should the first number of LSF times

be retrieved from the previous frame or are they in the bit-stream? For example, if the indicator bit is 1: the weighted LSF times are coded differentially to their predecessor in frame *k-1*, for frame *k* the first number of LSF times in the overlap area are derived from the LSFs in frame *k-1*. If the indicator bit is 0, the first LSF time of frame k is coded absolutely, all following LSFs are coded differentially to their predecessor.

In a practical embodiment, the LSF time frames are rather long, e.g. 1440 samples at 44.1kHz; in this case only around 30 bits per second are needed for this extra indication bit. Experiments showed that most of the frames could make use of the above technique advantageously, resulting in net bit savings per frame.

Further embodiment using overlapping frames

According to a further embodiment of the invention, the LSF time data is loss-lessly encoded. So instead of merging the overlap-pairs to single LSF times, the differences of the LSF times in a given frame are encoded with respect to the LSF times in another frame. So in the example of Figure 3 when the values l_0 until l_N are retrieved of frame *k-1*, the first three values l_0 until l_3 from frame *k* are retrieved by decoding the differences (in the bit-stream) to l_{N-2}, l_{N-1}, l_N of frame *k-1* respectively. By encoding an LSF time with reference to an LSF time in an other frame which is closer in time than any other LSF time in the other frame, a good exploitation of redundancy is obtained because times can best be encoded with reference to closest times. As their differences are usually rather small, they can be encoded quite efficiently by using a separate Huffman table. So apart from the bit denoting whether or not to use a technique as described in the first embodiment, for this particular example also the differences $l_{0,k} - l_{N-2,k-1}$, $l_{1,k} - l_{N-1,k-1}$, $l_{2,k} - l_{N,k-1}$ are placed in the bit-stream, in the case the first embodiment is not used for the overlap concerned.

Although less advantageously, it is alternatively possible to encode differences relative to other LSF times in the previous frame. For example, it is possible to only code the difference of the first LSF time of the subsequent frame relative to the last LSF time of the previous frame and then encode each subsequent LSF time in the subsequent frame relative to the preceding LSF time in the same frame, e.g. as follows: for frame *k-1*: $l_{N-1} - l_{N-2}$, $l_N - l_{N-1}$ and subsequently for frame *k*: $l_{0,k} - l_{N,k-1}$, $l_{1,k} - l_{0,k}$ etc.

System description

Fig. 7 shows a system according to an embodiment of the invention. The system comprises an apparatus 1 for transmitting or recording an encoded signal [S]. The

apparatus 1 comprises an input unit 10 for receiving at least part of an audio signal S, preferably a noise component of the audio signal. The input unit 10 may be an antenna, microphone, network connection, etc. The apparatus 1 further comprises an encoder 11 for encoding the signal S according to an above described embodiment of the invention (see in particular Figs. 4, 5 and 6) in order to obtain an encoded signal. It is possible that the input unit 10 receives a full audio signal and provides components thereof to other dedicated encoders. The encoded signal is furnished to an output unit 12 which transforms the encoded audio signal in a bit-stream [S] having a suitable format for transmission or storage via a transmission medium or storage medium 2. The system further comprises a receiver or reproduction apparatus 3 which receives the encoded signal [S] in an input unit 30. The input unit 30 furnishes the encoded signal [S] to the decoder 31. The decoder 31 decodes the encoded signal by performing a decoding process which is substantially an inverse operation of the encoding in the encoder 11 wherein a decoded signal S' is obtained which corresponds to the original signal S except for those parts which were lost during the encoding process. The decoder 31 furnishes the decoded signal S' to an output unit 32 that provides the decoded signal S'. The output unit 32 may be reproduction unit such as a speaker for reproducing the decoded signal S'. The output unit 32 may also be a transmitter for further transmitting the decoded signal S' for example over an in-home network, etc. In the case the signal S' is reconstruction of a component of the audio signal such as a noise component, then the output unit 32 may include combining means for combining the signal S' with other reconstructed components in order to provide a full audio signal.

Embodiments of the invention may be applied in, inter alia, Internet distribution, Solid State Audio, 3G terminals, GPRS and commercial successors thereof.

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims. In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. This word 'comprising' does not exclude the presence of other elements or steps than those listed in a claim. The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In a device claim enumerating several means, several of these means can be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

References

[1] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive sytems", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 309-321, June 1975.

[2] A.H. Gray, Jr. and J.D. Markel, "Quantization and bit allocation in speech processing", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 459-473, Dec. 1976.

[3] F.K. Soong and B.-H. Juang, "Line Spectrum Pair (LSP) and Speech Data Compression", Proc. ICASSP-84, Vol. 1, pp. 1.10.1-4, 1984.

[4] K.K. Paliwal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", IEEE Trans. on Speech and Audio Processing, Vol. 1, pp. 3-14, January 1993.

[5] F.K. Soong and B.-H. Juang, "Optimal Quantization of LSP Parameters", IEEE Trans. on Speech and Audio Processing, Vol. 1, pp. 15-24, January 1993.

[6] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals", J. Acoust. Soc. Am., 57, 535(A), 1975.

[7] N. Sagumura and F. Itakura, "Speech Data Compression by LSP Speech Analysis-Synthesis Technique", Trans. IECE '81/8, Vol. J 64-A, No. 8, pp. 599.606.

[8] P. Kabal and R.P. Ramachandran, "Computation of line spectral frequencies using chebyshev polynomials", IEEE Trans. on ASSP, vol. 34, no. 6, pp. 1419-1426, Dec. 1986.

[9] J. Rothweiler, "A rootfinding algorithm for line spectral frequencies", ICASSP-99.

[10] Engin Erzin and A. Enis Çetin, "Interframe Differential Vector Coding of Line Spectrum Frequencies", Proc. of the Int. Conf. on Acoustic, Speech and Signal Processing 1993 (ICASSP '93), Vol. II, pp.25-28, 27 April 1993